



# **COS 126 Exam Review Lectures 11-16**

## Special Announcement

---

In Friday's e-mail...

We invite you to join us at **The White House** on Friday December 11, 2015 for a *Workshop on K-12 Computer Science Education*. This convening will give stakeholders an opportunity to share their progress and to brainstorm new ways to expand computer science access even further. Participants will discuss their work and best practices on variety of topics including professional development, state & local polices, teaching frameworks, credentialing, and more. Senior administration officials are especially eager to hear your individual views on creative “blue sky” approaches focused on computer science learning outcomes, rather than just inputs.

Anything you think I should say?

*Send e-mail or stop by my office anytime Thursday.*



## REs

---

Q. How do you know what language is described by an RE?

Ex. Fall 2014 Question 5.

Let  $L = \{ \mathbf{ab}, \mathbf{aaab}, \mathbf{aaaab}, \mathbf{aabaab}, \mathbf{aabaaab} \}$ .

Write 1, 2, 3, or 4 to indicate whether the RE

1. Matches *no* strings in  $L$ .
2. Matches *only some strings in  $L$*  and some other strings.
3. Matches all strings in  $L$  and some other strings.
4. Matches *all strings in  $L$  and no other strings*.

$(aa^*b)^*$  3

$a^*b^*$  2

$(a|b)^*ab$  3

$a^*baba^*b^*$  1

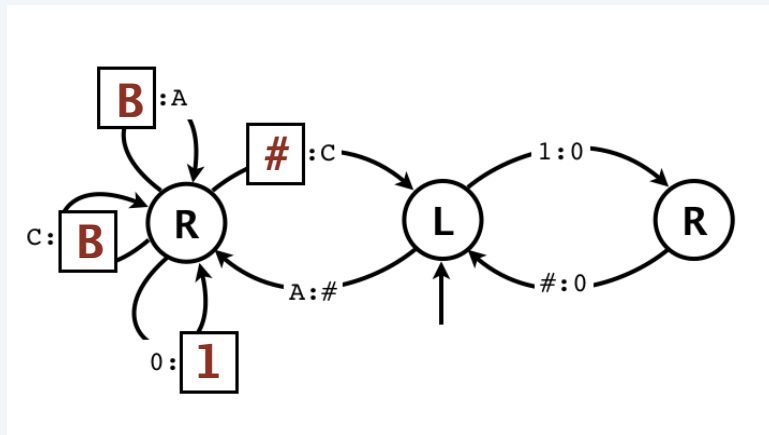
$(ab) | (a(a|aba)(a|aa)b)$  4

$a^*baaa^*b^*$  2

## Tracing TMs

Q. How do we trace a Turing Machine?

Ex. Fall 2014 Question 5.



# 1 # 1 A 1 1 B 1 1 1 **C** # # # # #

# 1 # 1 A 1 1 B 1 1 0 C **0** # # # # #

# 1 # 1 A 0 0 B 0 0 0 C 0 0 0 **0** 0 #

# 1 # 1 # **0** 0 B 0 0 0 C 0 0 0 0 0 #

# 1 # 1 # 1 1 A 1 1 1 B 1 1 1 1 1 **C** #

## Binary operations

---

Q. Why is  $\sim 0$  equal to  $-1$  and not  $1$ ? (Fall 2014 Q1B)

A (wrong).

$\sim$  is "not"

0 is "false"

"not false" is "true"

"true" is 1

A (correct).

$\sim$  is **BITWISE** "not"

0 is 00000000000000000000000000000000

$\sim 0$  is 11111111111111111111111111111111

11111111111111111111111111111111 is  $-1$  (2s complement)

## Poker hands

---

Q. How do you remove hands from the deck? (Fall 2014 Programming Exam)

**Part 1 (7 points).** Begin by downloading the files `Deck.java` and `FiveHands.txt` at

<http://www.cs.princeton.edu/~cos126/docs/data/Poker>

Your task now is to modify `Deck.java` to create a new program `RandomPokerHands.java` that (i) accepts an integer `N` from the command line, (ii) initializes a deck of cards, and (iii) performs the following operation `N` times: shuffle the deck and "deal" a five-card poker hand (by printing the first five cards in the deck). Print each hand on a separate line, using a two-character code for each card with C, D, H, S for Clubs, Diamonds, Hearts, and Spades (respectively) and T, J, Q, K, A for Ten, Jack, Queen, King, Ace (respectively). Note that you need to make sure that the suit and rank entries are one character each. For example, your program should produce output like the following:

```
% java-introcs RandomPokerHands 10
7H 3S TS QH 9D
2C 2S 7C 4C AS
JS 3S KD 7D 8H
5S 2C 2S 8H TS
QD JH KC 5C 4H
JC 7D 7S KC 8S
JC 3H AS 4C AD
3S TH KS 8H AH
AH 5C AC 5H 8C
5S 8S 7S TH 3C
```

A. You don't!

Advice.

READ CAREFULLY.

# 1D Cellular automata

Q. How does CA.java work? (Fall 2012 PE)

rules string

0 1 0 0 1 0 0 0

rules[]

0	000	0
1	010	1
2	010	0
3	011	0
4	100	1
5	101	0
6	110	0
7	111	0

input

0 0 1 1 1 0 1 0 0 1 0 0 0

one step

0 1 0 0 0 0 0 1 1 0 1 0 0

Key observation.

Rules come in as a *string*.

Easier to use an *array*.

```
// constructor saves the rules and sets initial state of each cell
public CA(int N, String bits) {
    // how many rules? Convert String to int[] for convenience
    int RNUM = bits.length();
    rules = new int[RNUM];
    for (int i = 0; i < RNUM; i++) {
        rules[i] = Integer.parseInt(bits.substring(i, i+1));
    }

    // 2N+1 cells plus dummy cell on each end
    cNum = 2*N + 3;
    // middle cell set to 1, rest are default zero
    cells = new int[cNum];
    cells[cNum/2] = 1;
}

// step method advances automaton one step
public void step() {
    // next set of cells
    int[] next = new int[cNum];

    // figure out the next state based on current cell and its neighbors
    // skip first and last cells which are dummy cells and always zero
    for (int i = 1; i < cNum - 1; i++) {
        // which rule? based on 4, 2, 1 binary place value
        int ruleIndex = cells[i-1]*4 + cells[i]*2 + cells[i+1];
        next[i] = rules[ruleIndex];
    }

    // transfer to current cells
    cells = next;
}
```

Advice. Read the comments (and write good comments).